

FIG. 1

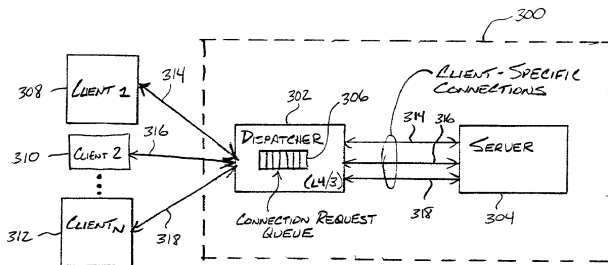


FIG. 3

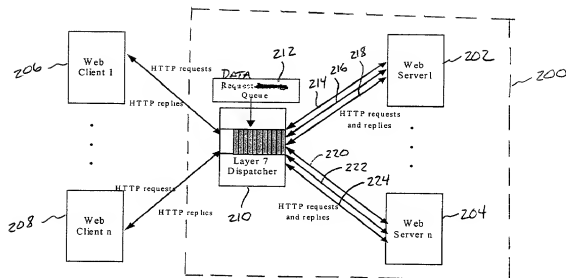


FIG. 2

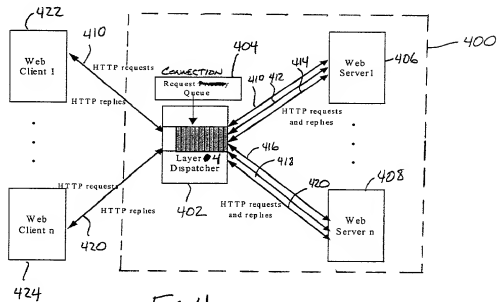


FIG. 4

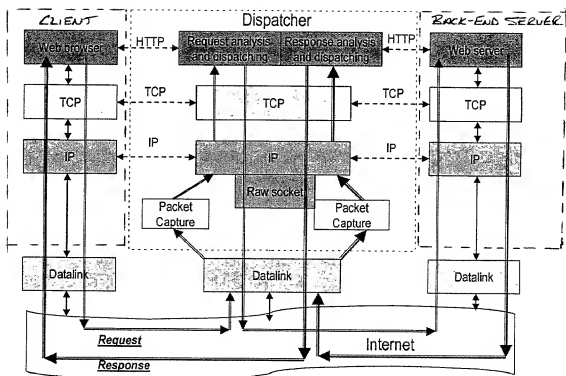


Figure 5

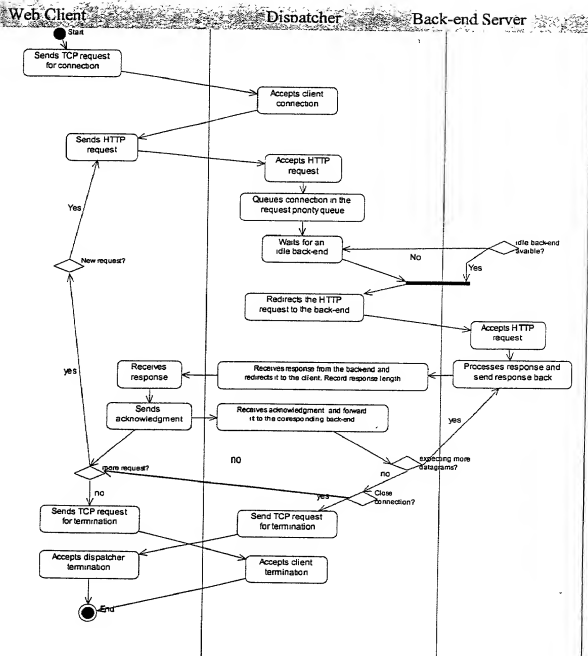


Figure 6

Client/Dispatcher

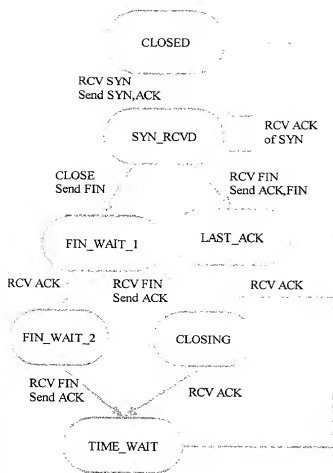


FIG. 7(a)

Dispatcher/Server

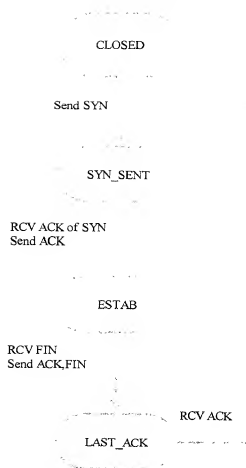


FIG. 7(b)

```
typedef struct{
    uint8_t flags;
    uint32_t collcount;
    uint16_t conn_priority;
    float domain_priority;
    enum net_state c_state;
    time_t start_time;
    uint32_t cIP;
    uint16_t cport;
    uint16_t dport;
    uint32_t next_send;
    uint32_t next_recv;
    enum req_state r_state;
    req *request;
    bool first_request;
};
```

c_mapping.

```
typedef struct{
    bool conn_close;
    uint32_t dtoc_start_seq;
    uint32_t c_start_seq;
    uint16_t req_num;
    float req_file_priority;
    int req_size;
    u_char *req_data;
    req *next;
};
```

req.

```
typedef struct{
    enum dport_state r_state;
    enum net_state s_state;
    uint32_t serv_start_seq;
    int response_length;
    uint32_t next_send;
    uint32_t temp_recv;
    uint32_t sIP;
    uint16_t sport;
    uint16_t dport;
    uint32_t hash_index;
};
```

s_mapping.

C_MAPPING: c_map[]
Mapping table for connections to clients

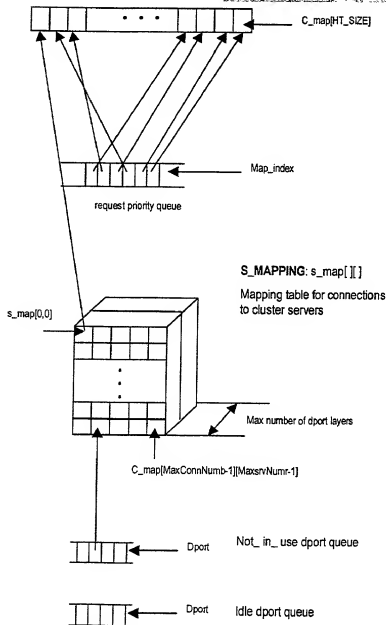


Figure 8

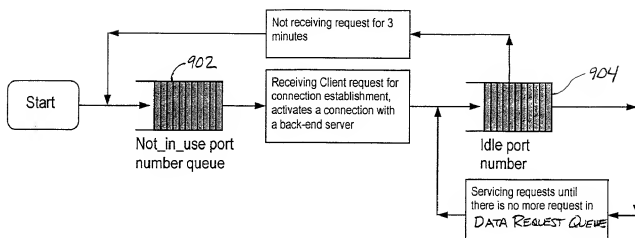


Figure 9

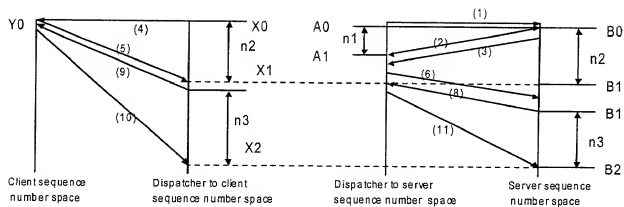


Figure 10